Trigger Accounting for 2002

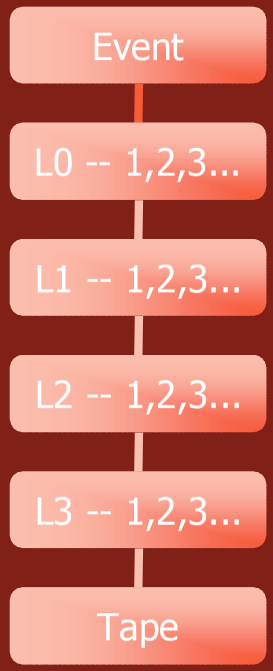
Jeff Landgraf

Issues to be addressed

- Multiple triggers at the same time
  - Mutually exclusive L0 Triggers
  - Interaction between L0/L1/L2/L3
- Offline interface
  - Scalers and counters
  - Evolution of trigger definitions

2001 STAR Trigger model

- Each event gets analyzed separately at each trigger level
- Events labeled by trgWord, noL3Bias()
- Only worked because:
  - No L0 overlap attempted
  - No L1/L2 used



L0 Overlap Issue

Example: Mixed Central (ps=1) & Min-bias (ps=100)	
L0 Trigger	**Trigger word, PS**
not Central && not Min-bias	N/A
not Central && Min-bias	1, 100
Central && not Min-bias	2, 1
Central && Min-bias	3, 1 (conflict: MB → 100, Central → 1)

- Using the trigger word alone gives a biased Min-bias
- Reconstructing an unbiased trigger is simple:
  - Min-bias → every TW=1 and 1 of every 100 TW=3
  - Central → every TW=2 and every TW=3
- L1 marks each event according to this rule → "L1 rescaling."
- The Configuration/L1 rescale algorithm works with arbitrary triggers

Interaction between L0/L1/L2/L3

- Correlated triggers (analyzed naively) introduce bias
- To untangle these interactions we would need:
  - $N_{L0} * 2^{N_{L1}} * 2^{N_{L2}} * 2^{N_{L3}}$ counters
  - Additional PS logic (Accept untriggered)
  - Very complicated analysis logic
- Instead, I will show how to avoid the problem entirely by some simple constraints on how L1/L2 & L3 algorithms are defined.

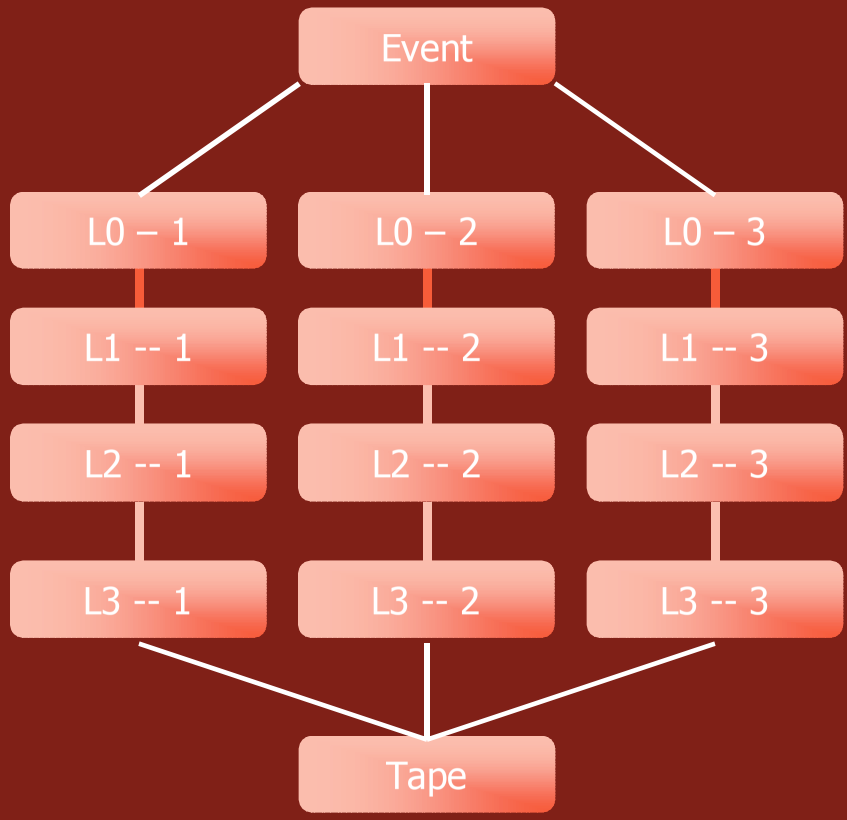
example:

L2: High Pt L3: High multiplicity	→	High Pt, High Multiplicity events

2002 STAR Trigger Model

- Pretend that we have N independent trigger systems.
- Each system has its own L0, L1, L2 and L3 components.
- Configure one trigger on each system.
- If any one of the trigger systems accepts the trigger, the event is saved to tape.
- The event gets marked according to which trigger systems accepted it.
- No accounting biases.

→ 2002 we simulate this situation



Implementation of the 2002 Trigger Model

- Configure N Triggers in run control. The configuration for each trigger is roughly equivalent to setting last years TRG_SETUP parameters.
- Every Trigger has a L0 requirement and exactly one algorithm at each level, L1, L2 & L3.
- Then, as we run:
  - Perform L1 rescaling before running any L1 algorithms.
    - Result is that each event is labeled according to which Triggers were satisfied at L0.
  - The L1, L2, & L3 algorithms for each Trigger check to ensure that the Trigger was satisfied at the previous level. If not, the event is ignored by that algorithm.
- These two steps are all that is needed to ensure that no bias is introduced by running multiple triggers.

Scalers and Counters

- Some scaler information will be stored to the database. The information will be organized by Trigger. The following will be available for each Trigger:
  - The number of events satisfying the physics of the L0 component without regard to the detector busy
  - The number of events seen and rejected by the L1, L2 and L3 components
  - The prescales for each trigger at each level
- These counters will be written every 2 minutes or so during the run. The contributions from different trigger levels will be synchronized to ~1 second.
- This is enough information to obtain absolute cross sections in units of (bunch crossings) $^{-1}$

**note: This is a VERY small part of the information available in the scaler boards (5 boards * 2^24 counters.)

Offline Event Labeling

- In the data file each trigger is represented by a bit in a 32-bit mask, the TriggerID. The value of this bit is arbitrary.
- The database contains the key to translate the TriggerId into a meaningful identifier.
- Jerome has made a request for service work for someone to provide the interface that reads the database to make the scalers and the trigger descriptions easily accessible offline.

Evolution of Trigger Definitions

- The trigger word has many disadvantages for use as the offline event selection criteria:
  - A given word has different meanings in different configurations (vertexMinBias, MinBiasVertex, etc...)
  - It can only specify one trigger, even though the event can satisfy more than one trigger
  - It is insensitive to threshold changes
  - It is insensitive to TCU bit definition changes
  - It is insensitive to PS changes
- The offline trigger identifier will replace the trigger word for event selection.
- It will have one field that describes the trigger. This field will be fixed for all time to the same value.
- It will also have separate version fields for:
  - Trigger Definition (TCU Lookup tables & Tier 1 files are the same)
  - Threshold Values (All thresholds must be the same)
  - P.S. (The full set of prescales must be the same)
- These versions can be obtained automatically from the configuration files each run and stored to a database. We will provide a tool to browse this database and check the documentation for each version.